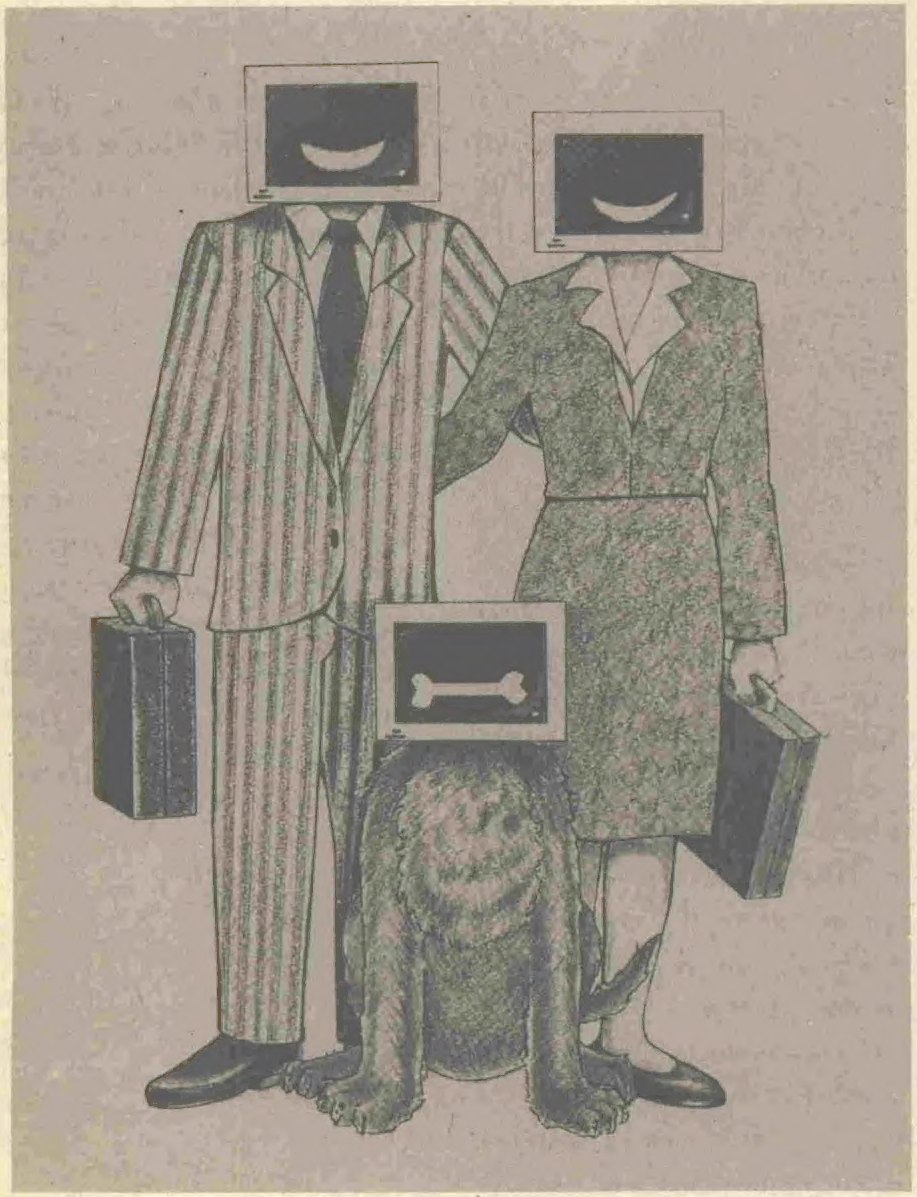
บทความพิเศษ

WORD PROCESSORS

สหัส พรหมสิทธิ์

เวิร์ดโปรเซสซิ่ง
แบบหลากภาษา

ความพยายามของมนุษย์ในอันที่จะทลายอุปสรรคกำแพงภาษาที่ขวางกั้นใกล้เคียงความจริงแล้ว



ความหมายของคอมพิวเตอร์ที่เป็นของดั้งเดิมที่สุดนั้นน่าจะได้ชื่อว่าเป็นจักรกลที่ออกแบบขึ้นมาให้ทำหน้าที่เป็นเครื่องคำนวณ โดยเฉพาะในแง่ของการเป็น mathematics machine แต่ถ้าจะมองเข้าไปให้ลึกอีกนิดก็จะเห็นว่า อานุภาพขั้นพื้นฐานของคอมพิวเตอร์ไม่ได้อยู่ที่การคำนวณ แต่จะมีขีดความสามารถที่หนักไปในแง่ของการสามารถยักย้ายถ่ายเทสัญลักษณ์อักษรต่าง ๆ (linguistic tokens) เพื่อให้เกิดความเหมาะสมด้วยการกำกับความหมายให้สัญลักษณ์ดังกล่าว สรรพนามใหม่ของคอมพิวเตอร์จึงน่าจะได้ชื่อว่าเป็นจักรกลแห่งภาษาหรือ language machine เมื่อเป็นเช่นนี้จึงยืนยันต่อไปได้ว่า ศูนย์กลางอันเป็นแก่นของวิทยาศาสตร์คอมพิวเตอร์ (computer science) ก็คือการประมวลผลในสิ่งที่เรียกว่าภาษาชาวบ้าน (natural language) อันหมายถึงภาษาที่มนุษย์ใช้พูดและเขียน ซึ่งจะตรงข้ามกับภาษาประดิษฐ์ (artificial languages) ที่ใช้สำหรับสร้างคอมพิวเตอร์โปรแกรม สิ่งที่น่าสนใจก็คือว่าบทบาทของคอมพิวเตอร์ในเรื่องของภาษาชาวบ้านนั้นมีมาตั้งแต่ช่วงหลังปี 1950 อันเป็นช่วงแรก ๆ ที่ยังเจริญรอยตามแนวความคิดที่ได้มาจากกิจกรรมทางทหาร กิจกรรมเช่นว่านี้ก็คือประสบการณ์ในการใช้คอมพิวเตอร์เพื่อถอดรหัสที่ใช้ในทางทหารระหว่างสงครามโลกครั้งที่สอง จนกระทั่งถึงปี 1949

WORD PROCESSORS

เวิร์ดโปรเซสซิ่งแบบหลากภาษา

อันเป็นการก้าวมาถึงจุดหนึ่งที่ได้นำคอมพิวเตอร์แบบดิจิตอลมาใช้ในกิจกรรมด้านทหารอย่างเต็มที่ และได้มีนักวิทยาศาสตร์ชื่อวอร์เร็น วีเวอร์ (Warren Weaver) แถลงว่า เทคนิคต่าง ๆ ที่พัฒนาขึ้นมาเพื่อแกะหรือถอดรหัสนั้นน่าจะนำไปใช้ในการแปลภาษาด้วยคอมพิวเตอร์ได้ ซึ่งในเวลาต่อมาก็ได้มีการพิสูจน์ว่าไม่เป็นความจริง เพราะภาษาชาวบ้านหรือ natural language มีความดิดดิ้นมากกว่ารหัสหลายท่านัก อย่างไรก็ตาม ความพยายามใช้คอมพิวเตอร์ในช่วงหลังปี 1950 เพื่อให้แปลเท็กซ์ (text) จากภาษาหนึ่งไปสู่อีกภาษาหนึ่งนั้นได้ก่อให้เกิดการรุดหน้าในด้านนี้มากพอสมควร แม้ว่าจะยังห่างไกลจากเป้าหมายที่จะต้องบรรลุอยู่ตาม แต่การค้นคว้าในด้านนี้ยังคงดำเนินต่อไปในทิศทางที่ไม่สู้ชัดเจนนักนอกจากสิ่งที่จะทำให้ภาษาชาวบ้านเป็นสื่อกลางสำหรับติดต่อสื่อความกับคอมพิวเตอร์ให้ได้

ใช้คอมพิวเตอร์แปลภาษา

ความใฝ่ฝันอันสูงสุดของนักวิทยาศาสตร์คอมพิวเตอร์ในแง่ของภาษาพูดและภาษาเขียนในยุคปัจจุบันที่น่าจะเป็นการก้าวล้ำจากที่กล่าวในตอนต้น ๆ (หลังปี 1950) ก็เห็นจะเป็นเรื่องของนักวิจัยกลุ่มหนึ่งที่พยายามจะพัฒนาทฤษฎีแบบรวมยอดเบ็ดเสร็จ (unified theories) ในด้านการคำนวณซึ่งจะรวบรวมทั้งภาษาชาวบ้านและภาษาประดิษฐ์เข้ามาไว้ในที่เดียวกัน แต่ในที่นี้จะให้ความสนใจเฉพาะประเด็นของภาษาที่มนุษย์ใช้สื่อความกันในชีวิตประจำวันเท่านั้น ซึ่งในแง่นี้ก็ได้เกิดซอฟต์แวร์มากมาย และในจำนวนนี้ก็มีทั้งซอฟต์แวร์ที่เป็นแบบพื้นฐานธรรมดาและใช้ได้ดี เช่น พวกเวิร์ดโปรเซสเซอร์ จนถึงแบบที่ซับซ้อนมาก ๆ ในโลกของการค้นคว้าเรื่องภาษากับคอมพิวเตอร์ส่งผลให้เกิดคอมพิวเตอร์ซอฟต์แวร์ที่ใช้ในการประมวลผลที่เป็นภาษาชาวบ้านหลายประเภท ซึ่งถ้าจะให้จำแนกอย่างกว้าง ๆ ก็จะมีอยู่ 4 ประเภทด้วยกันคือ

1. โปรแกรมหรือซอฟต์แวร์ที่ใช้สำหรับแปลภาษามนุษย์
2. โปรแกรมที่ใช้ในเวิร์ดโปรเซสซิ่ง
3. โปรแกรมสำหรับการตอบคำถามต่าง ๆ
4. โปรแกรมที่ทำหน้าที่ในแง่ของการสื่อความในรูปของระบบประสานเชื่อมโยง (coordination systems) ที่เรียกว่าอิเล็กทรอนิกส์เมล์หรืออี-เมล์

ในแต่ละกรณีดังกล่าวอาศัยการวิเคราะห์ลักษณะของความเหมาะสมทางภาษา (linguistic competence) รวมทั้งความสัมพันธ์ระหว่างความเหมาะสมกับโครงสร้างของกฎที่เป็นทางการ (formal rule structures) ที่ถือว่าเป็นรากฐานทางทฤษฎีของซอฟต์แวร์ทั้งหมด ซึ่งก่อนที่จะพูดถึงแนวความคิดในด้านเวิร์ดโปรเซสซิ่งนั้นผู้เขียนจะเล่าสิ่งที่เกี่ยวข้องที่เห็นว่าน่าสนใจ เพื่อเป็นการปูทางและให้ผู้อ่านมีพื้นฐานคร่าว ๆ ว่า หลักเกณฑ์ที่จะทำให้คอมพิวเตอร์เข้าใจภาษามนุษย์นั้นต้องมีอะไรบ้าง โดยจะเริ่มที่เรื่องของการแปล เพื่อให้ผู้อ่านมองเห็นปัญหาต่าง ๆ ที่จำเป็นสำหรับทำความเข้าใจสิ่งที่กล่าวถึงในภายหลัง

การแปลจากภาษาหนึ่งไปสู่อีกภาษาหนึ่งด้วยคอมพิวเตอร์ในยุคแรก ๆ นั้นจะเป็นการค้นคว้าที่เน้นหนักทางเทคนิคของการบรรจุพจนานุกรมขนาดใหญ่เข้าไปในคอมพิวเตอร์สตอเรจ แล้วหาทางทำให้คอมพิวเตอร์รู้จักค้นหาความหมายของศัพท์ต่าง ๆ จากพจนานุกรมอย่างมีประสิทธิภาพ ส่วนซอฟต์แวร์ที่จะเข้ามาทำหน้าที่ในด้านไวยากรณ์ก็จะอาศัยทฤษฎีที่เกี่ยวกับโครงสร้างของภาษา พร้อมด้วยกฎเกณฑ์ต่าง ๆ ที่จำเป็นสำหรับการแปลภาษา กฎเกณฑ์ที่ว่านี้ได้มาจากประสบการณ์ในรูปของ rough-and-ready rules ส่วนวิธีการแปลด้วยคอมพิวเตอร์เท่าที่เคยคิดกันในอดีตเมื่อยุคแรก ๆ นั้นเข้าใจว่าไม่น่าจะยากเย็นอะไร คือถ้าคิดจะแปลจากภาษาหนึ่งไปเป็นอีกภาษาหนึ่งก็จำเป็นต้องรู้คำศัพท์หรือความหมายของแต่ละคำในแต่ละประโยคเป็นอันดับแรก เสร็จแล้วก็แปลทีละคำ จากนั้นทำการเรียงลำดับของถ้อยคำเพื่อให้เกิดความถูกต้องชัดเจนในแต่ละประโยคเป็นอันว่าจบสิ้น แต่ตามความเป็นจริงแล้วจะไม่ง่ายดังที่คิด เพราะการแปลแบบนี้อาจได้ผลลัพธ์ออกมาเลวมากจนอ่านไม่เข้าใจ สาเหตุนี้เองที่การแปลภาษาจะแตกต่างกับการถอดรหัสด้วยคอมพิวเตอร์ เพราะความหมายต่าง ๆ ที่แฝงในภาษามนุษย์นั้นจะต่างกับความหมายในรหัสอย่างมากในแง่ที่ว่าความหมายของแต่ละประโยคของภาษามนุษย์นอกจากจะขึ้นกับลักษณะของประโยคแล้ว ยังขึ้นกับเนื้อหาสาระที่มีอยู่ในประโยคนั้น ๆ รวมทั้งอาจจะเกี่ยวข้องกับประโยคอื่น ๆ อีกด้วย ซึ่งทั้งหมดนี้เรียกว่าเกิดจากความกำกวม (ambiquity) มีความหมายสองนัยหรืออาจมากกว่านั้น นอกจากนี้ก็ยังแบ่งความกำกวมดังกล่าวออกไปอีก เช่น lexical ambiquity, structural ambiquity, ambiquity of deep structure, semantic ambiquity, pragmatic ambiquity

WORD PROCESSORS
เวิร์ดโปรเซสซิ่งแบบหลากภาษา

ความหมายของเวิร์ดโปรเซสซิ่ง

เวิร์ดโปรเซสซิ่งหรือเท็กซ์โปรเซสซิ่ง (text processing) ก็คืออออฟฟิศออโตเมชั่นฟาซิลิตี้อย่างหนึ่งที่ทำให้ผู้ใช้สามารถจัดการเอกสารสิ่งพิมพ์ต่าง ๆ โดยอาศัยคอมพิวเตอร์พร้อมด้วยเครื่องช่วยอื่น ๆ เพื่อเรียบเรียง แก้ไข ดัดต่อ (edit) รวมทั้งการรีฟอร์แมต (reformat) และเก็บสิ่งที่จะพิมพ์ ตลอดจนการพิมพ์เป็นเอกสารหรือรูปเล่ม ซึ่งทั้งหมดนี้คอมพิวเตอร์มีส่วนช่วยให้เป็นไปด้วยความยืดหยุ่นและคล่องตัวที่สุด ซึ่งถ้าจะพูดถึงประเด็นสำคัญสำหรับเรื่องนี้ก็เห็นจะหนีไม่พ้นตัวซอฟต์แวร์ที่ทำหน้าที่ในด้านเวิร์ดโปรเซสซิ่งที่เรียกว่าเวิร์ดโปรเซสเซอร์

การทำงานของเวิร์ดโปรเซสเซอร์จะเน้นหนักในแง่ของการยักย้ายถ่ายเทและดิสเพลย์สัญลักษณ์อักษรต่าง ๆ ที่อยู่ในรูปของ strings of characters โดยจะเกี่ยวข้องกับเรื่องโครงสร้างของภาษาไม่มากนัก แต่สำหรับในหลาย ๆ กรณีอาจสร้างปัญหามากพอสมควร ปัญหาดังกล่าวมักเป็นเรื่องของเทคนิคที่ถือว่าเป็นแก่นของการออกแบบคอมพิวเตอร์ซอฟต์แวร์ อนึ่ง ในบางกรณีนั้นผลิตภัณฑ์ที่เป็นเวิร์ดโปรเซสซิ่งโปรแกรมจะให้รูปของบรรทัดที่ประกอบกันเป็นเท็กซ์ ส่วนในกรณีอื่นเวิร์ดโปรเซสซิ่งโปรแกรมหรือเวิร์ดโปรเซสเซอร์จะเข้าไปเกี่ยวข้องกับความยุ่งยากซับซ้อนของเลย์เอาต์ (layout) ตัวพิมพ์ต่าง ๆ ตลอดจนการแทรกรูปภาพ

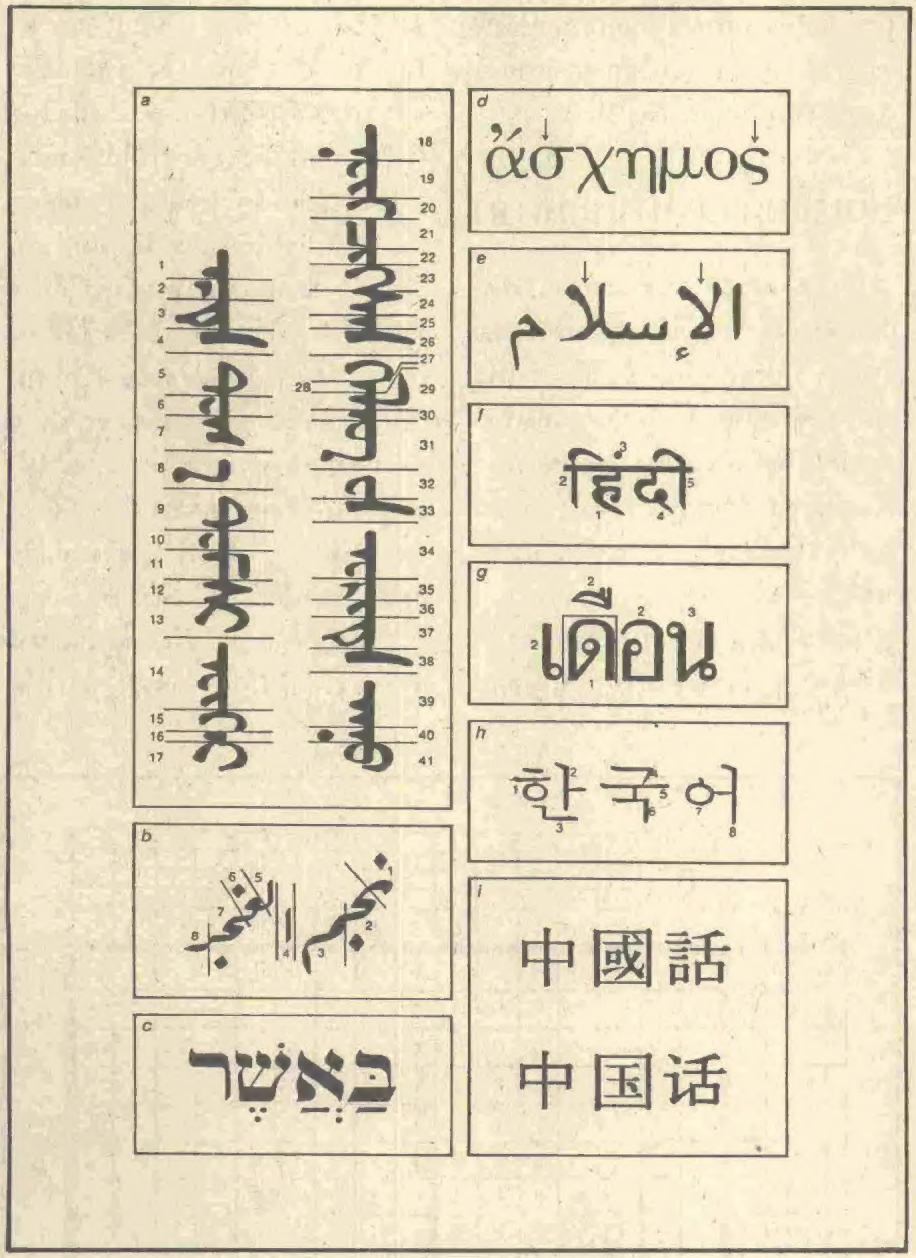
กุญแจดอกสำคัญในการออกแบบเวิร์ดโปรเซสซิ่งซอฟต์แวร์ก็คือ เรื่องของ representation กับอินเตอร์แอคชั่น (interaction) หน้าที่ของ representation ก็คือการกำหนดดาต้าสตรัคเจอร์ (data structures) ที่ซอฟต์แวร์จะเข้าไปยักย้ายถ่ายเทหรือจัดการได้อย่างสะดวก รวมทั้งการมองเผื่อในเรื่องอื่น ๆ เช่น เลย์เอาต์ของสิ่งพิมพ์ ส่วนอินเตอร์แอคชั่นจะเกี่ยวกับวิธีการออกคำสั่งต่าง ๆ ของผู้ใช้และการโต้ตอบของระบบเวิร์ดโปรเซสซิ่ง

ทุกครั้งที่กล่าวถึงดาต้าสตรัคเจอร์มักจะมีเรื่องของอัลกอริทึม (algorithms) เข้ามาเกี่ยวข้องด้วยเสมอ ดังนั้นทั้งดาต้าสตรัคเจอร์กับอัลกอริทึมจึงเปรียบเหมือนวัตถุดิบหรือสิ่งที่นำมาใช้ในการสร้างคอมพิวเตอร์โปรแกรม เพราะแม้แต่ในตัวของคอมพิวเตอร์เองก็ไม่ได้ประกอบด้วยอะไรอื่นนอกจากดาต้าสตรัคเจอร์กับอัลกอริทึม ดาต้าสตรัคเจอร์สำหรับในกรณีนี้ก็คือรีจิสเตอร์กับเมมโมรีเวิร์ด (registers and memory words) ที่มีอยู่ภายในตั้งแต่แรกและใช้เป็นที่เก็บค่าตัวเลขแบบไบนารี (binary values) ในทำนองเดียวกันอัลกอริทึมที่อยู่ในรูปของฮาร์ดแวายร์ (hardwired algorithms) จะประกอบด้วยกฎต่าง ๆ ที่ตายตัว (fixed rules) และแฝงในวงจรอิเล็กทรอนิกส์ลอจิกสำหรับอินเตอร์พรีต (interprete) ดาต้าให้อยู่ในรูปของอินสตรัคชันสำหรับเอ็กซีคิวต์ (execute) ต่อไป ดังจะเห็นได้จากขั้นพื้นฐานที่สุดว่า คอมพิวเตอร์สามารถประมวลผลได้เฉพาะข้อมูลหรือดาต้าเพียงแบบเดียว นั่นคือ ดาต้าที่อยู่ในรูปของบิต (bits) หรือไบนารีดิจิต (binary digits) ดังในรูปที่ 1

คอมพิวเตอร์นั้นจะเป็นเสมือนดีไวซ์หรือจักรกลที่มีความเป็นอเนกประสงค์อย่างที่สุดในแง่ของการทำงานที่จะฝากทุกสิ่งทุกอย่างไว้กับโปรแกรม ในแง่ที่ว่าข่าวสารข้อมูลเหมือนสายน้ำที่ประมวลผลโดยอาศัยโปรแกรมหนึ่ง

TYPE	DISPLAYED FORM	INTERNAL REPRESENTATION
CARDINAL	83	00000000 00000000 00000000 01010011 (MAGNITUDE)
INTEGER	−83	1 1111111 11111111 11111111 10101101 (SIGN; MAGNITUDE)
REAL	83.0	0 1000111 10100110 00000000 00000000 00000000 00000000 00000000 00000000 (SIGN OF MANTISSA; WEIGHTED EXPONENT; NORMALIZED MANTISSA)
SET	0,1,4,6	00000000 00000000 00000000 01010011 (MEMBERS OF SET)
CHARACTER	S	01010011 (ASCII CODE)
STRING	SET 83	S E T <SPACE> 8 3 00000110 01010011 01000101 01010100 00100000 00111000 00110011 (NUMBER OF CHARACTERS; ASCII CODES OF CHARACTERS)

รูปที่ 1 แสดงดาต้าประเภทต่าง ๆ พร้อมด้วยการดิสเพลย์ให้เห็น พร้อมลักษณะของการเก็บดาต้าเช่นนี้ในรูปของไบนารีดิจิตในคอมพิวเตอร์



รูปที่ 2 ตัวอย่างของภาษาต่าง ๆ ที่ใช้ในโลกนี้และได้มีการนำเข้าสู่ระบบเวิร์ดโปรเซสซิ่ง เช่น ภาษามองโกเลียน (a) หมายถึง "ในกาลครั้งหนึ่งนานมาแล้ว....." ถัดไปเป็นภาษาอาหรับ (b) ที่แปลว่า "ดวงดาวแห่งรุ่งอรุณ" ข้ายล่างสุด (c) เป็นภาษาฮีบรูที่แปลว่า "ที่ไหน" (d) เป็นภาษากรีกที่มีความหมายว่า "น่าเกลียด" (e) เป็นภาษาอิสลามหมายถึง "อิสลาม" (f) เป็นภาษาฮินดี (Hindi) และอ่านว่า hindi (h) เป็นภาษาเกาหลีที่มีคำแปลเหมือนอย่างที่เรียกนั่นคือ "ภาษาเกาหลี" และ (i) เป็นภาษาจีนแปลว่า "ภาษาจีน"

ในขณะนั้นสามารถแปรสภาพจากข้อมูลไปอยู่ในรูปของโปรแกรมได้ด้วยตัวเอง หมายความว่าคอมพิวเตอร์โปรแกรมนั้นถูกสร้างขึ้นมาจากความคิดอ่านที่คนส่วนใหญ่สามารถเข้าใจเพื่อทำหน้าที่อย่างใดอย่างหนึ่งเสร็จแล้วจะมีโปรแกรมที่เรียกว่าแอสเซมเบลอร์ (assembler) หรือคอมพายเลอร์ (compiler) ทำหน้าที่แมพ (map) ความคิดอ่านดังกล่าวให้อยู่ในฟาซิลิตี้ต่าง ๆ ในคอมพิวเตอร์ ความเข้าใจเช่นนี้จะเป็นหนทางให้สามารถสร้างระบบอะไรก็ได้ไม่ว่าระบบนั้น ๆ จะยุ่งยากซับซ้อนสักแค่ไหน ในแง่ที่ว่าโปรแกรมเมอร์จะเป็นผู้กำหนดลำดับของนามธรรม (hierarchy of abstractions) โดยเริ่มจากการมองภาพกว้างที่สุดแล้วให้ความสนใจส่วนต่าง ๆ ที่ประกอบกันขึ้นเป็นภาพทีละส่วน ก่อนก้าวสู่รายละเอียดของแต่ละส่วนในภายหลัง

ปรัชญาของเวิร์ดโปรเซสซิ่ง

โลกของเวิร์ดโปรเซสซิ่งในรูปแบบที่กว้างที่สุดนั้นจะต้องอยู่ในรูปของการประมวลผลได้ทุกภาษาเท่าที่ใช้กันในปัจจุบัน ดังรูปที่ 2 ส่วนวิธีที่จะออกแบบซอฟต์แวร์ให้ทำเช่นนี้ได้ก็จะต้องเริ่มต้นด้วยความคิดที่เป็นหลักใหญ่ ๆ เกี่ยวกับความเป็นไปได้ 3 ประการคือ

1. ความเป็นไปได้ในการเก็บ (represent) เท็กซ์ในคอมพิวเตอร์

2. ความเป็นไปได้ในการพิมพ์เท็กซ์โดยอาศัยคีย์บอร์ดของคอมพิวเตอร์

3. ความเป็นไปได้สำหรับคอมพิวเตอร์ที่จะทำหน้าที่เป็นสื่อในการนำเท็กซ์ส่งให้ผู้พิมพ์

หน้าที่ของทั้งสามหัวข้อข้างต้นพอสรุปได้ดังนี้

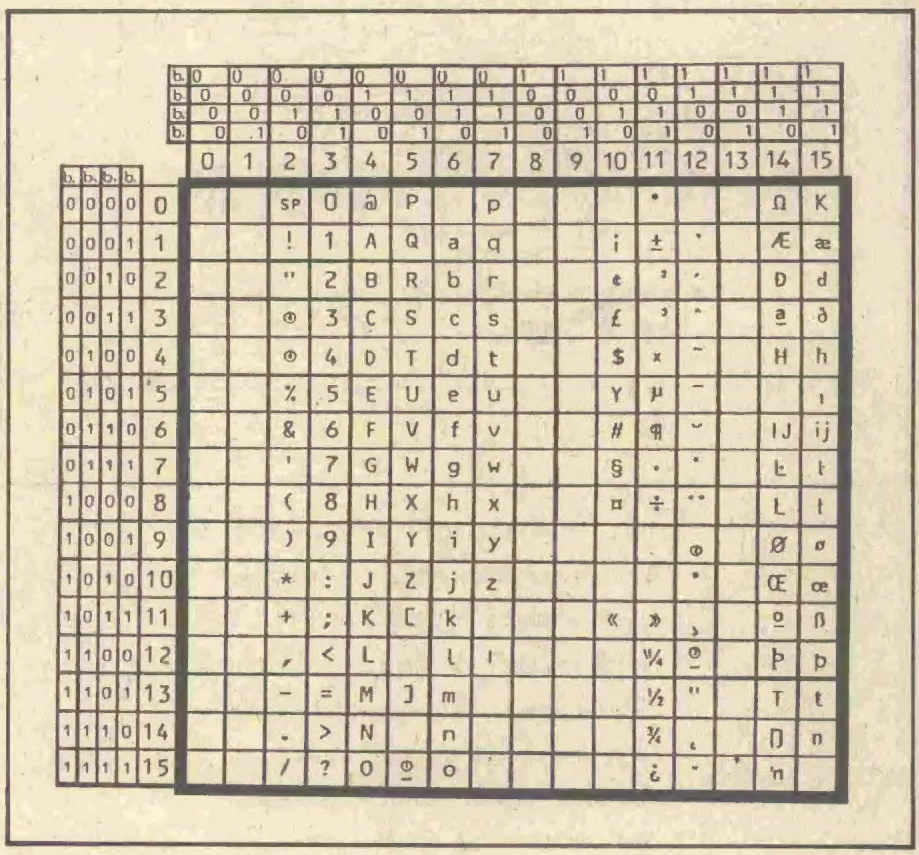
1. เอ็นโค้ดดิ้ง (encoding)
2. การพิมพ์ (typing)
3. เร็นเดอร์ริ่ง (rendering)

เอ็นโค้ดดิ้งหมายถึงการเก็บตัวเลขข้อมูลต่าง ๆ ไว้ในคอมพิวเตอร์ในรูปของไบนารีดิจิต (หรือบิต) หรือไบนารีโค้ดนัมเบอร์ ซึ่งในกรณีของภาษาอังกฤษที่ใช้ระบบโค้ดแอสกีจะมีรายละเอียดดังในรูปที่ 3 สำหรับกรณีที่มีการเก็บข้อมูลแบบ 8 บิตต่อหนึ่งไบต์ หมายถึงมี 256 หนทางที่จะเก็บตัวเลข ตัวอักษร และสัญลักษณ์ในรูปของไบต์ที่ประกอบด้วยเลขศูนย์และเลขหนึ่งจำนวน 8 ตัว จึงไม่เพียงพอสำหรับภาษาอื่น ส่วนเร็นเดอร์ริ่งหมายถึงการดิสเพลย์สิ่งที่เป็นเท็กซ์บนจอภาพของคอมพิวเตอร์ รวมทั้งการพิมพ์เท็กซ์ลงบนกระดาษ

การเก็บข้อมูลในคอมพิวเตอร์ในรูป 8 บิตต่อไบต์มีข้อจำกัดมากเพราะเก็บสัญลักษณ์ต่าง ๆ ได้เพียง 256 แบบเท่านั้น จึงจำเป็นต้องหาวิธีแก้ไขข้อบกพร่องนี้ หากต้องการจะทำให้เกิดระบบเวิร์ดโปรเซสซิ่งแบบหลากภาษา (multilingual word processing) ที่สามารถประมวลผลเกี่ยวกับภาษาหลาย ๆ ภาษาดังตัวอย่างในรูปที่ 2 กับของจริงที่แสดงไว้ในรูปที่ 4 อันเป็นส่วนหนึ่งของ screen dump ที่แสดงสิ่งที่มีอยู่ในดิสเพลย์สกรีนของคอมพิวเตอร์ ซึ่งเป็นผลิตภัณฑ์ของบริษัท Xerox ส่วนในตัวผลิตภัณฑ์จะประกอบด้วยเวิร์ดโปรเซสซิ่งซอฟต์แวร์แบบหลากภาษา เช่น โรมัน รัสเซีย กรีก ญี่ปุ่น (Hiragana, Katakana, Kanji) จีน (Hanzi, Bopomofo) เกาหลี (Hangul) อาหรับ และฮีบรู พีซีเวอร์กสเตชันที่เรียกว่า Star ทีมงานผู้พัฒนาเรื่องนี้ของ Xerox มีความเห็นว่า ประโยชน์สูงสุดของคอมพิวเตอร์ในด้านเวิร์ดโปรเซสซิ่งนั้นน่าจะทำให้รู้จักภาษามากแบบที่สุดเท่าที่จะทำได้ และต้องยอมให้ภาษาหลาย ๆ ภาษาปะปนกันด้วย ดังรูปที่ 4

เอ็นโค้ดดิ้งสำหรับทุกภาษา

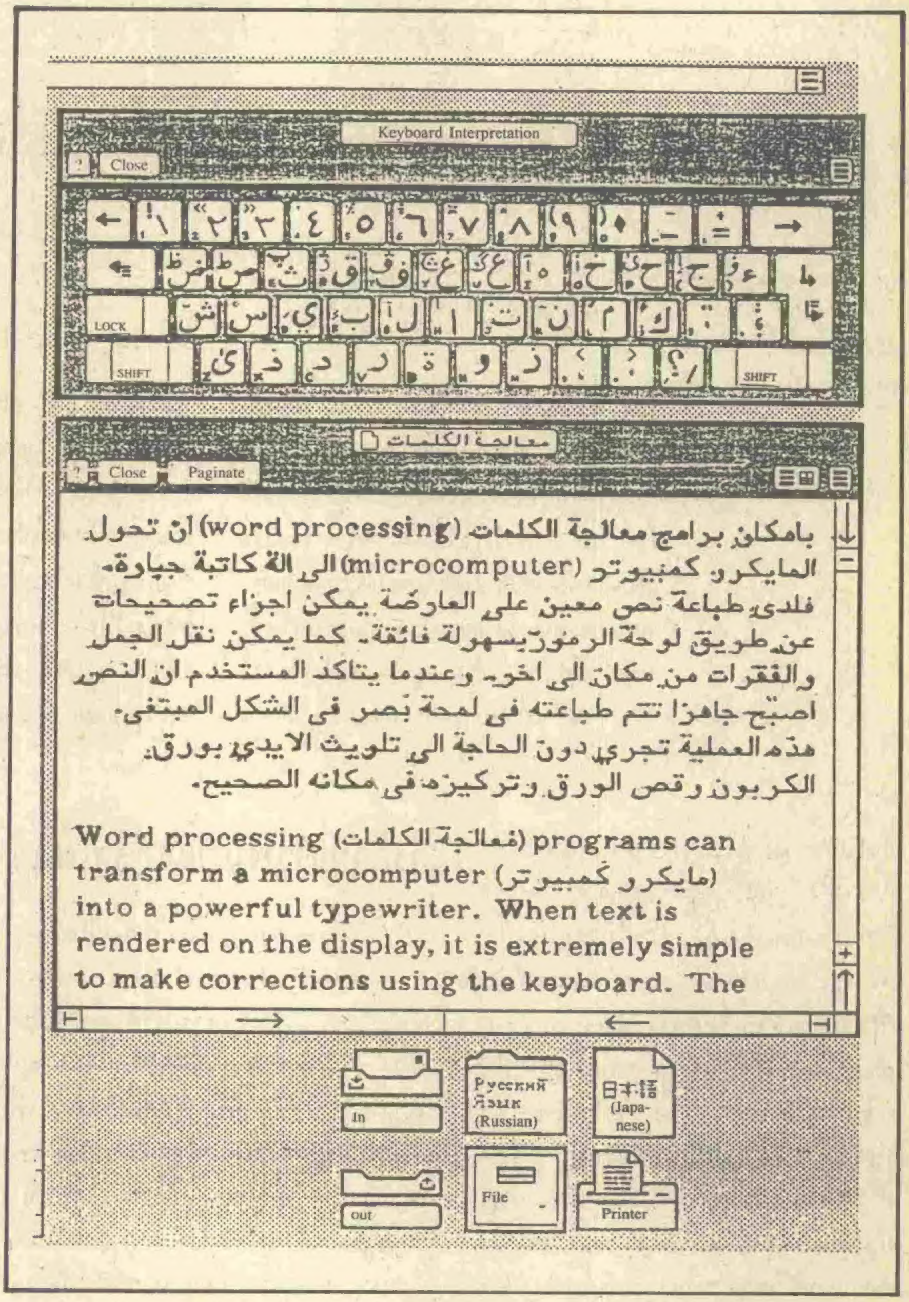
ได้กล่าวไว้แล้วว่างานด้านเวิร์ดโปรเซสซิ่งสามารถจำแนกเป็นกิจกรรมใหญ่ ๆ 3 ประการคือ เอ็นโค้ดดิ้ง การพิมพ์และเร็นเดอร์ริ่ง โดยได้พูดถึงเรื่องของเอ็นโค้ดดิ้งไว้บ้างแล้วในแง่ของการเก็บข้อมูลลงสู่คอมพิวเตอร์ในรูป 8 บิตต่อหนึ่งไบต์จะเก็บข้อมูลได้เท่ากับ 2^8 หรือ 256 แบบ แต่ภาษาอีกหลาย ๆ ภาษาในโลก (นอกเหนือจากภาษาอังกฤษ) จะมีจำนวนตัวอักษร ตลอดจนสัญลักษณ์อื่น ๆ มากกว่า 256 ตัว (ดูรูปที่ 3 ประกอบ) จึงไม่พอเพียงที่จะทำการเอ็นโค้ดโดยใช้แค่หนึ่งไบต์ (หรือ 8 บิต) แต่จะต้องใช้โค้ดดิ้งสกีม (coding scheme) แบบสองไบต์ติดต่อกัน เพื่อทำให้มีจำนวนโค้ดได้เท่ากับ 2^{16} หรือ 65,536 หรือในบางครั้งก็อาจต้องใช้จำนวนโค้ดเท่ากับ 2^{24} หรือมากกว่า 16 ล้านโค้ดจึงจะพอ อย่างไรก็ตามเอ็นโค้ดดิ้งที่แทนอักษรแต่ละตัวด้วยความยาวสองหรือสามไบต์นั้นจะสิ้นเปลืองเนื้อที่ในหน่วยความจำของคอมพิวเตอร์มาก จึงต้องหาวิธีเอ็นโค้ดดิ้งแบบยืดหยุ่นได้ (flexible encoding) สำหรับเลือกว่า เมื่อไรจึงจะขยายความยาวของโค้ดดิ้งสกีมออกเป็นสองไบต์



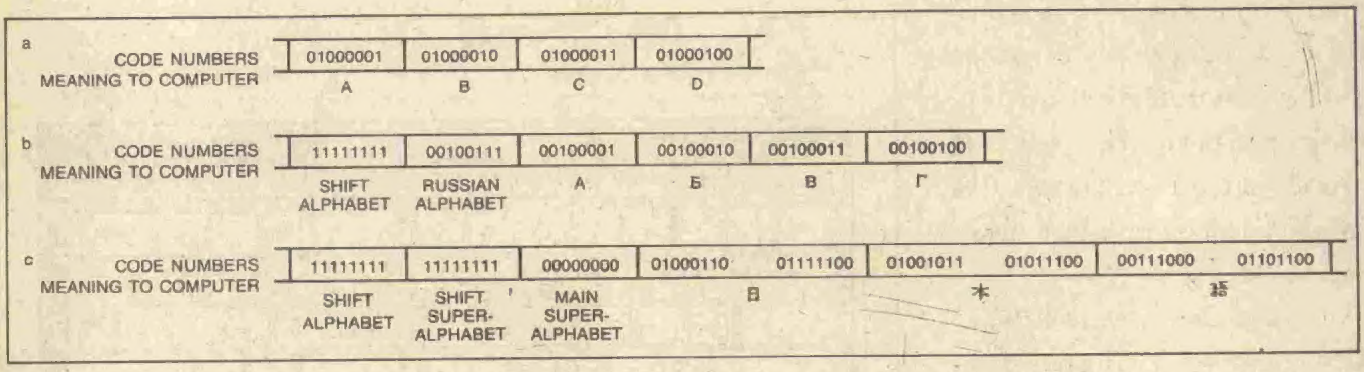
รูปที่ 3 แสดงการเก็บตัวอักษรและสัญลักษณ์ต่าง ๆ ในคอมพิวเตอร์โดยใช้ระบบโค้ดของแอสกี

หรือสามไบต์ต่ออักษรหนึ่งตัว ซึ่งก็ทำได้ด้วยการกำหนดให้มีจำนวนสองหรือสามไบต์ในตอนแรกเริ่มของการเอ็นโค้ดดิ้ง เพื่อส่งสัญญาณให้คอมพิวเตอร์รู้ว่ากำลังจะเข้าสู่ภาษาของชาติใด โดยถือว่าสัญญาณดังกล่าวจะเป็นส่วนหนึ่งของ encode text

งานขั้นแรกสำหรับการเอ็นโค้ดสิ่งที่เรียกว่าเทิกซ์สำหรับหลาย ๆ ภาษา (multilingual text encoding) นั้นจะเป็นเรื่องการกำหนดไบนารีโค้ดนัมเบอร์สำหรับแต่ละภาษาในโลก เช่น ภาษารัสเซียจะมีโค้ดนัมเบอร์เท่ากับ 00100111 ภาษาโรมันมีโค้ดนัมเบอร์เป็น 00000000 ส่วนภาษากรีกและภาษาอาหรับโค้ดนัมเบอร์เท่ากับ 00100110 และ 11100000 เป็นต้น การตั้งรหัสสำหรับอักษรแต่ละตัวของแต่ละภาษาจะอาศัยข้อกำหนดของมาตรฐานสากล ขั้นต่อไปเป็นการกำหนดสัญญาณที่เรียกว่า shift alphabet เพื่อบอกให้คอมพิวเตอร์รู้ว่าภาษาที่กำลังจะพูดถึงนั้นไม่ใช่ภาษาอังกฤษ โดยอาศัยโค้ดที่เป็นตัวเลขเท่ากับ 111-11111 ผู้อ่านสามารถทำความเข้าใจได้จากรูปที่ 5 ส่วนในกรณีของภาษาที่มีความซับซ้อนมาก ๆ (เช่น ภาษาจีน) จำนวนรูปแบบของตัวอักษรมากกว่า 65,536 แบบ จึงต้องใช้จำนวนโค้ดในระดับที่สูงกว่า 2^{16} อันอาจต้องใช้รหัสแบบสามไบต์ (three-byte scheme) ที่ยอมให้จำนวนโค้ดได้มากถึง 2^{24} หรือกว่า 16 ล้าน ในกรณีเช่นนี้สิ่งที่จะต้องทำเป็นอันดับแรกคือ การแบ่งกลุ่มของตัวอักษรให้เป็นกลุ่มย่อย ๆ ขนาดกลุ่มละ 65,536 ตัวอักษรในรูปของ superalphabet ที่อยู่ถัดจาก shift alphabet ซึ่งจะมีรหัสเท่ากับ 11111111 เหมือนกัน อันจะเห็นได้จากในภาพ c. ของรูปที่ 5 ที่เริ่มต้นด้วย shift alphabet บ่งบอก



รูปที่ 4 แสดงเวอร์ชวลคีย์บอร์ด (virtual keyboard) ของอักษรแบบอารบิก (Arabic) ที่ทำงานร่วมกับคอมพิวเตอร์ โดยตอนกลางของภาพแสดงเอกสารที่มีข้อความทั้งภาษาอาหรับและอังกฤษผสมกัน ภาษาอาหรับนั้นจะอ่านจากขวาไปซ้าย ส่วนภาษาอังกฤษจะอ่านตามปกติ ในกรณีนี้คอมพิวเตอร์จะช่วยให้ผู้พิมพ์สามารถสอดแทรกภาษาหนึ่งเข้ากับอีกภาษาหนึ่ง แต่ยังคงรักษาทิศทางในการอ่านของแต่ละภาษาให้ถูกต้องเหมือนเดิมทุกอย่าง

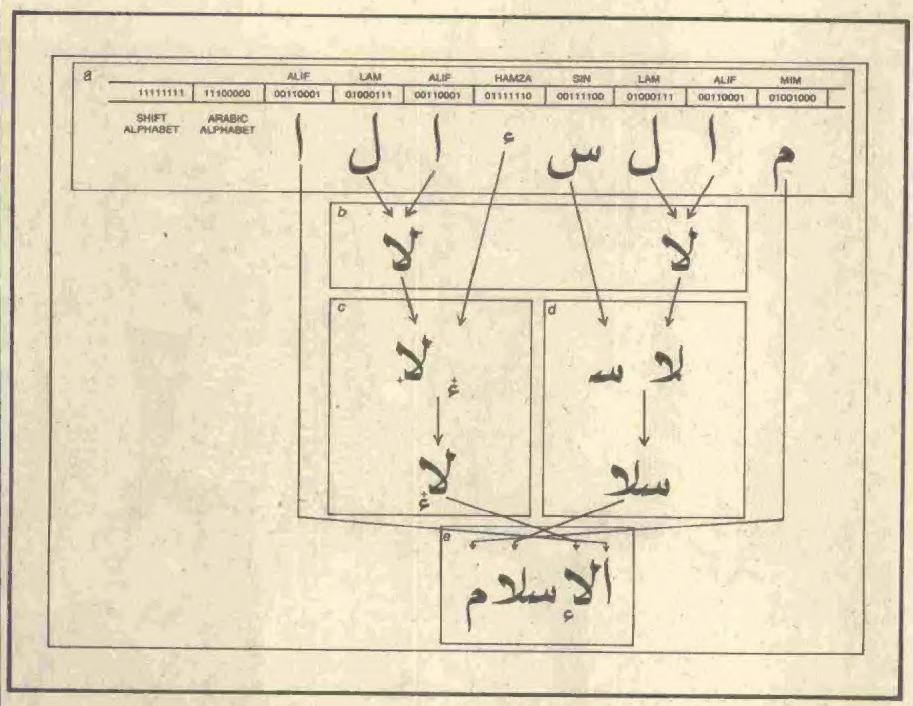


รูปที่ 5 แสดงการทำงานของ flexible encoding ที่สามารถนำทุกภาษาที่มีใช้อยู่ในโลกมาทำเป็นเวิร์ดโปรเซสซิ่งในรูปของการประหยัดจำนวนไบต์ให้มากที่สุด เช่น (a) ซึ่งถือว่าเป็นสกีมมาตรฐาน (standard scheme) ที่ใช้แทนภาษาอังกฤษและใช้ระบบโค้ดของแอสกีกำหนดให้แทนอักษรแต่ละตัวด้วยความยาวหนึ่งไบต์ (หรือ 8 บิต) ส่วนภาษาอื่นจะขึ้นต้นด้วยสิ่งที่เรียกว่า shift alphabet เช่นในกรณีของ (b) เป็นการเอ็นโค้ดภาษารัสเซีย shift alphabet ส่งสัญญาณบอกให้คอมพิวเตอร์รู้ว่าเริ่มเข้าสู่ตัวอักษรที่ไม่ใช่ภาษาอังกฤษ ส่วนไบต์ถัดมานั้นจะใช้รหัส 00100111 เพื่อแจ้งให้คอมพิวเตอร์ทราบว่าเป็นภาษารัสเซีย สำหรับภาษาจีนซึ่งถือว่าซับซ้อนมาก ๆ นั้นก็จะมีรูปแบบของเอ็นโค้ดดิ้งดังแสดงไว้ใน (c) คือจะเริ่มด้วย shift alphabet ที่มีรหัสเป็น 11111111 เสร็จแล้วตามด้วยสัญญาณที่เป็น shift superalphabet และ main superalphabet (00000000)

ว่าไม่ใช่ภาษาอังกฤษ ส่วนไบต์ที่สองก็จะเป็น shift superalphabet แสดงถึงการแบ่งกลุ่มของตัวอักษรออกเป็นกลุ่มละ 65,536 ตัว และถัดจากนี้จะเป็นไบต์ที่สามเรียกว่า main superalphabet แสดงด้วยรหัส 00000000 เป็นการเข้าสู่ภาษาจีน เพราะในตำแหน่งที่อยู่ถัดไปนั้นจะกินเนื้อที่ระดับสองไบต์สำหรับแต่ละตัวอักษร ยกเว้นอักษรที่มีรูปแบบยากเกินหรือไม่ค่อยได้ใช้ จะต้องแทนด้วยรหัสที่ยาวมากกว่า 2 ไบต์ ภาษาที่จัดว่าซับซ้อนในข่ายเดียวกับภาษาจีนก็คือภาษาญี่ปุ่น ทั้งสองภาษาต้องการสองไบต์แรกที่มีรหัสเท่ากับ 11111111 ส่งสัญญาณที่เป็น shift superalphabet ส่วนไบต์ที่สามจะบอกประเภทของ superalphabet ที่ใช้ ซึ่งจะตามด้วยตัวอักษรที่มีขนาดความยาวเรียงกันตัวละสองไบต์ เพื่อแสดงลักษณะเฉพาะของภาษานั้น ๆ (ideographs)

ว่าด้วยการพิมพ์และดิสเพลย์

ในแง่ของการพิมพ์ภาษาเกือบทุกภาษาที่มีอยู่ในโลกสามารถใช้คีย์บอร์ดของพิมพ์ดีดมาตรฐานมาดัดแปลงให้เข้ากันได้ เพราะการพิมพ์ที่อาศัยคอมพิวเตอร์นั้นจะมีซอฟต์แวร์คอยทำหน้าที่เปลี่ยนแปลงการอินเตอร์พรีต (interpretation) คีย์ต่าง ๆ ที่เกิดจากคอมพิวเตอร์ เพื่อให้คีย์ที่เคยทำหน้าที่พิมพ์อักษรในภาษาหนึ่งไปพิมพ์ตัวอักษรของอีกภาษาหนึ่งได้ เช่น คีย์ที่แทนอักษร A ในภาษาอังกฤษสามารถแทนตัว I ในภาษารัสเซียหรือแทน س ในภาษาอาหรับ ขึ้นกับ keyboard interpretation นอกจากนี้คอมพิวเตอร์ยังสามารถดิสเพลย์ไดอะแกรมเล็ก ๆ ที่เรียกว่า virtual keyboard ให้ปรากฏบนสกรีนดังรูปที่ 4 (ภาพบนสุด) เพื่อบอกผู้พิมพ์ว่าในขณะนั้นได้กำหนดให้คีย์บอร์ดแทนอักษรของภาษาอะไรและคีย์ไหนแทนตัวอักษรอะไรบ้าง ซึ่งสำหรับบางภาษาคอมพิวเตอร์จะช่วยให้การพิมพ์ง่ายขึ้นดังรูปที่ 6 เช่นในกรณีของ special combination character ที่แทนคำว่า lam กับ alif ทุกครั้งที่เห็นสองคำนี้เรียงกัน สำหรับพิมพ์ดีดภาษาอาหรับพวกตัว special character จะมีคีย์เฉพาะของแต่ละอักษร สำหรับ lam กับ alif (ดูรูปที่ 6) คอมพิวเตอร์สามารถรวมการทำงานของอักขระทุกตัวให้เหลือเพียง special character เพียงตัวเดียว ทำให้ประหยัดการใช้คีย์ได้มาก นอกจากนี้คอมพิวเตอร์ยังสามารถประมวลผลตัวอักษรต่าง ๆ ที่มีรูปลักษณ์แตกต่างกันของเท็กซ์ได้อย่างไม่เดือดร้อน เพราะเวิร์ดโปรเซสซิ่งซอฟต์แวร์จะช่วยจัดการกับความยุ่งยากซับซ้อนของสคริปต์ (script) โดยผู้พิมพ์จะทำหน้าที่แค่เพียงป้อนคาแรกเตอร์ต่าง ๆ เข้าไป (a na-



รูปที่ 6 แสดงกระบวนการในการดิสเพลย์เท็กซ์บนสกรีนหรือการพิมพ์ลงบนกระดาษด้วยการยกตัวอย่างที่เป็นภาษาอาหรับ เพื่อแสดงคำที่อ่านว่า al-islam ซึ่งในภาพ (a) เป็นการเก็บคำดังกล่าวในรูปของอักษรที่เรียงกันแปดตัวที่จะต้องมีการคำนวณแบบพิเศษ เพื่อใช้อักษรดังกล่าวมาสร้างเป็นคำ โดยในขั้นแรกนั้นคอมพิวเตอร์จะมุ่งไปที่ตัวอักษรที่เกี่ยวข้องและอยู่ใกล้กันเพื่อรวมเป็นคำควบให้เหลือเพียงตัวเดียว (a single lingature) เช่นคำว่า lam กับ alif เมื่อรวมกันแล้วจะมีลักษณะคล้ายอักษร gamma ในภาษากรีก (b) คอมพิวเตอร์จะเก็บข้อสังเกตเช่นนี้ไว้ในหน่วยความจำชั่วคราว โดยยังคงรักษาตัวอักษรดั้งเดิมต่อไป คอมพิวเตอร์จะมองหาเครื่องหมายเล็ก ๆ ที่เรียกว่า hamza ซึ่งอยู่ตรงส่วนบนหรือส่วนล่างของตัวอักษร (c) โดยเครื่องหมาย + ที่ปรากฏนั้นจะทำหน้าที่เป็น registration marks คอมพิวเตอร์จะอาศัยขบวนการแบบเดียวกันนี้ในการรวมคำว่า sin กับ lam-alif เข้าด้วยกัน จนเกิด special joining form ดังแสดงไว้ใน (d) ขั้นสุดท้ายคอมพิวเตอร์จะแสดงคำหรือพิมพ์คำดังกล่าวลงในกระดาษโดยเรียงลำดับจากขวาไปซ้าย (e)

tural or phonetic sequence of characters) อย่างไรก็ตาม บางภาษามีสคริปต์ที่ยุ่งยากซับซ้อน (เช่น ภาษาจีน ญี่ปุ่น และในบางลักษณะของภาษาเกาหลี) ที่ท้าทายความสามารถในด้านการพิมพ์ของคอมพิวเตอร์ และต้องการการค้นคว้าพัฒนาอีกพอสมควรก่อนที่ทุกอย่างจะสะดวกขึ้น

เร็นเดอร์ริ่งเท็กซ์ที่เก็บอยู่ในคอมพิวเตอร์ ในแง่ของภาษาอังกฤษนั้นจะมีความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างโค้ดนัมเบอร์ในคอมพิวเตอร์สตอเรจกับคาแรกเตอร์ที่อยู่ในสภาพที่จะนำมาดิสเพลย์บนสกรีนหรือนำมาพิมพ์บนกระดาษ แต่ในกรณีที่ใช้หลายภาษาพร้อม ๆ กัน (multilingual text) จำเป็นต้องเลิกใช้ความสัมพันธ์แบบหนึ่งต่อหนึ่ง แล้วเปลี่ยนไปใช้สกีมที่มีความยืดหยุ่นคล่องตัวมากขึ้น เนื่องจากในขณะที่คอมพิวเตอร์ประมวลผลสิ่งที่เรียงลำดับเป็นแถวเป็นแนวในรูปของเท็กซ์โค้ด เพื่อแสดงบนสกรีนหรืองานพิมพ์นั้น สามารถคำนวณหารูปแบบการเรียงลำดับของโค้ดให้ครบทุกรูปแบบในแต่ละการคำนวณ รวมทั้ง combination ต่าง ๆ ได้อย่างสมบูรณ์เพื่อใช้ในการเลือกกราฟิคัลฟอร์มที่มีหลายแบบ (graphical forms) และตำแหน่งของคาแรกเตอร์ว่าอยู่ตรงไหนในประโยค นอกจากนี้ยังต้องป้อนสิ่งที่เรียกว่า variant character forms เข้าไปในคอมพิวเตอร์สำหรับใช้ในกรณีที่ตัวอักษรมีการเปลี่ยนแปลงรูปอันเกิดจากการใช้ภาษาหลาย ๆ แบบ ตัวอย่างในทางปฏิบัติที่จะต้องอาศัยแนวความคิดที่กล่าวถึงทั้งหมดนั้นสามารถนำมาแสดงให้เห็นได้ เช่น อักษรของภาษาอาหรับที่เปลี่ยนแปรได้ถึงสี่แบบ (forms) ขึ้นกับว่าอักษรนั้นอยู่เดี่ยว ๆ ในตอนต้น

T	01010100	T	01010100
h	01101000	h	01101000
e	01100101	e	01100101
	00100000		00100000
w	01110111	w	01110111
o	01101111	o	01101111
r	01110010	r	01110010
d	01100100	d	01100100
s	01110011	s	01110011
	00100000		00100000
	11111111		11111111
	11100000		11100000
ا	00110001	ا	00110001
ل	01000111	ل	01000111
ا	00110001	ا	00110001
ع	01111110	ع	01111110
ر	00111100	ر	00111100
ل	01000111	ل	01000111
ا	00110001	ا	00110001
ر	01001000	ر	01001000
	11111111		00110000
	00000000	ب	01001011
	00100000		00110000
a	01100001	ا	00110001
n	01101110	ل	01000111
d	01100100	ع	01000010
	00100000	ر	00111010
	11111111	ب	00110010
	11100000		11111111
ا	00110001		00000000
ل	01000111		00100000
ع	01000010	m	01101101
ر	00111010	e	01100101
ب	00110010	a	01100001
	11111111	n	01101110
	00000000		00100000
	00100000	ا	01001001
m	01101101	s	01110011
e	01100101	ا	01101100
a	01100001	a	01100001
n	01101110	m	01101101
	00100000		
ا	01001001		
s	01110011		
ا	01101100		
a	01100001		
m	01101101		

รูปที่ 7 แสดงการเก็บเท็กซ์สองภาษาในคอมพิวเตอร์ ประกอบด้วยประโยคที่ว่า "The words al-islam and al' arab mean Islam and the Arabs" โดยเก็บคำต่าง ๆ ที่เป็นอารบิกไว้ในรูปของภาษาอาหรับ และจะเก็บประโยคไว้ในคอมพิวเตอร์ในรูปที่เรียกว่า phonetic order ดังซีกซ้ายของรูป ส่วนทางซีกขวาจะแสดงสภาพที่ประโยคดังกล่าวผ่านการ edit โดยแทนคำว่า and ด้วยเครื่องหมายที่เป็นภาษาอาหรับ ซึ่งจะต้องลบคำว่า a, n และ d ตลอดจน shift codes ต่าง ๆ ออกให้หมด แล้วแทรกโค้ดที่เป็นตัวอักษรอาหรับที่อ่านว่า wa เข้าไปแทน

ตอนกลางหรือตอนท้ายของคำ ฯลฯ จึงจำเป็นต้องอาศัยทั้งกฎเกณฑ์ ตลอดจนแนวคิดทุกแง่ทุกประเด็นที่เกี่ยวกับการใช้ภาษาแต่ละแบบ แล้วสอนให้คอมพิวเตอร์ทำตามกฎเกณฑ์ รวมทั้งปลูกฝังเพื่อให้สามารถนึกคิดแบบรอบด้าน

ผู้เขียนจะแสดงให้ผู้อ่านเห็นว่าในแง่ของการเก็บภาษาอักษรในคอมพิวเตอร์ ตลอดจนการแทรกสิ่งต่าง ๆ เข้าไปในประโยคที่ต้องใช้ภาษามากกว่าหนึ่งภาษาขึ้นไปนั้นเขาทำกันอย่างไร ซึ่งถ้าพิจารณาตามรูปที่ 7 ก็จะเข้าใจได้อย่างชัดเจนว่า การเก็บเท็กซ์สองภาษา เช่น อังกฤษกับอาหรับ เมื่ออยู่ในสตอเรจของคอมพิวเตอร์แล้วจะมีลักษณะเช่นใด เมื่อมีการแทรกอะไรเข้าไปในประโยคแล้วจะเกิดอะไรขึ้น

อนาคตของเวิร์ดโปรเซสซิ่งแบบหลากภาษา

จุดมุ่งหมายในการใช้งานขั้นแรกสำหรับเวิร์ดโปรเซสซิ่งแบบหลากภาษาเกิดจากการติดต่อสื่อความระหว่างประเทศที่มีความแตกต่างกันในเรื่องของภาษานั้นนับวันจะทวีความสำคัญขึ้นเรื่อย ๆ จนจำเป็นต้องเกี่ยวข้องกับเอกสารแบบหลากภาษา (multilingual documents) ส่วนโครงสร้างขั้นพื้นฐานที่ทำให้การติดต่อสื่อสารแบบนี้เป็นไปได้ก็จะต้องอาศัยเน็ตเวอร์กของคอมพิวเตอร์เวอร์กสเตชันที่มีอยู่ทั่วโลกทำหน้าที่รับและส่งข่าวสารข้อมูล รวมทั้งเท็กซ์และเอกสารต่าง ๆ ด้วยวิธีทางอิเล็กทรอนิกส์ เช่น อิเล็กทรอนิกส์เมล์หรืออี-เมล์ ดังตัวอย่างของการค้าข้ามชาติที่ทำธุรกิจเกี่ยวกับน้ำมัน โดยสมมติว่าสำนักงานใหญ่ของบริษัทนี้ตั้งอยู่ที่มลรัฐเท็กซัส

ในสหรัฐฯ และเป็นหนึ่งในบรรดาบริษัทใหญ่ ๆ ของโลกที่มีเน็ตเวอร์กอยู่ทั่วโลก ซึ่งในเวลาปฏิบัติงานประจำวันนั้นจะมีเอกสารที่เป็นภาษาอาหรับส่งมาจากประเทศในเอเชียตะวันออกกลางโดยวิธีทางอิเล็กทรอนิกส์เข้ามายังคอมพิวเตอร์ที่ตั้งอยู่ในมลรัฐเท็กซัส แต่ยังไม่มีซอฟต์แวร์ที่จะมาจัดการกับอารบิกเท็กซ์ คอมพิวเตอร์ในมลรัฐเท็กซัสมีขีดความสามารถที่จะรับรู้อะไรหลาย ๆ อย่างในแง่ของเท็กซ์หรือเอกสาร ในอันที่จะไม่ปล่อยให้ทุกอย่างพังครืนลงไป เพราะเกิดความเข้าใจผิดว่าโค้ดที่เป็นภาษาอาหรับ (Arabic codes) นั้นเป็นภาษาอังกฤษ โดยอย่างน้อยที่สุดก็ต้องสามารถเก็บงำข้อความในส่วนที่เป็นภาษาอังกฤษ (ของเท็กซ์แบบหลากภาษา) เอาไว้ให้ได้ รวมทั้งสามารถรันซอฟต์แวร์โมดูลที่เป็นภาษาอาหรับได้ด้วย ฯลฯ

ตัวอย่างของการค้าและการติดต่อระหว่างประเทศในเรื่องของน้ำมันดังกล่าวข้างต้นจะเป็นต้นกำเนิดในเรื่องของการออกแบบที่จะต้องมองภาพให้กว้างที่สุดสำหรับระบบเวิร์ดโปรเซสซิ่งที่มีหลาย ๆ ภาษาภายในนั้น โดยให้ความสนใจหลักการ 3 ข้อคือ

1. compatibility เป็นเครื่องบังคับว่าคอมพิวเตอร์ทุกตัวที่อยู่ในระบบจะต้องสามารถรับส่งและแลกเปลี่ยนเอกสารข้อความต่าง ๆ ที่มีรูปแบบของตัวอักษรอะไรก็ได้

2. open-endedness ที่บังคับว่าคอมพิวเตอร์ทุกตัวจะต้องสามารถรับรู้ชนิดต่าง ๆ ของตัวอักษรทุกภาษาของโลกที่จะเป็นอะไรก็ได้

3. modularity ที่กำหนดให้คอมพิวเตอร์สามารถรับรู้ภาษาใหม่ ๆ ได้ทีละภาษาโดยเพิ่มเข้าไปได้คราวละหนึ่งภาษา

สรุปท้าย

เป้าหมายทั้งสามประการยังเป็นของยากอยู่สำหรับเมื่อราว 5-6 ปีก่อน แต่ไม่น่าจะเกินความสามารถของมนุษย์ตราบเท่าที่ผู้ออกแบบระบบเวิร์ดโปรเซสซิ่งมองสิ่งที่เป็นเท็กซ์ให้กว้างกว่าที่เป็นอยู่พอที่จะครอบคลุมภาษาต่าง ๆ ที่ใช้ในโลกทุกวันนี้ได้หมด อย่างไรก็ตาม สิ่งที่จะเข้ามาเกี่ยวข้องกับเวิร์ดโปรเซสซิ่งแบบหลากภาษาอย่างมากในอนาคตอันใกล้นี้เห็นจะเป็นอิเล็กทรอนิกส์เมล์ที่ทำหน้าที่รับส่งข้อความและเอกสารต่าง ๆ ได้เร็วมาก โดยอาศัยอิเล็กทรอนิกส์เน็ตเวอร์กทั้งที่เป็นของส่วนบุคคล จนถึงคอมมิวนิเคชันไลน์สาธารณะ ซึ่งเป็นเรื่องที่เป็นไปได้ เพราะในขณะนี้ระบบอิเล็กทรอนิกส์เมล์ที่เป็นแบบสากล (international electronic-mail systems) นั้นก็มีใช้กันอยู่แล้ว

เรื่องของระบบอิเล็กทรอนิกส์เมล์เหมือนกับกิจการไปรษณีย์และโทรศัพท์ในแง่ที่ว่า ประโยชน์ของการใช้สอยจะแปรตามความกว้างไกลที่เน็ตเวอร์กจะสามารถคลุมถึง จึงค่อนข้างมั่นใจว่าสักวันหนึ่งการใช้บริการในด้านอิเล็กทรอนิกส์เมล์จะเหมือนกับโทรศัพท์ที่ทำหน้าที่เป็นสาธารณูปโภคที่มีอยู่ทั่วโลก อันจะเป็นประโยชน์สำคัญสำหรับการใช้ระบบเวิร์ดโปรเซสซิ่งที่ผนึกรวมนานาภาษาสากลไว้ภายใน และจะเป็นการใช้งานร่วมกับระบบอิเล็กทรอนิกส์เมล์แบบที่มีเน็ตเวอร์กอยู่ทั่วโลกที่ติดต่อสื่อความกันโดยใช้ภาษาหลาย ๆ แบบ